🏢 Jirani App - Estate Management Platform

Show Image

Show Image

Show Image

A comprehensive digital ecosystem that connects estate residents, management, service providers, and property owners through a secure and user-friendly interface.

Show Image

🎯 Project Overview

Jirani App is a web-based estate management system built with Django that serves as the estate's central command center — automating billing, streamlining support, simplifying communication, and enabling smart living for modern communities.

Built by: [Your Name]
Institution: SCITT
Framework: Django 5.2
Development Period: 5 weeks

✨ Features

✅ Implemented Features

- **User Authentication** - Secure login/logout for residents, agents, and landlords

- **Resident Dashboard** - Real-time view of payments, requests, and community info
- **Payment Tracking** - Monitor rent, additional services, maintenance fees, and debt
- **Request Management** - Submit and track maintenance/service requests
- **Work Order System** - Track work orders with status and priority indicators
- **Unit Management** - View vacant and occupied units
- **Responsive Design** - Works on desktop, tablet, and mobile devices
- **Modern UI/UX** - Clean, intuitive interface with gradient branding

🚧 Planned Features (Future Development)

- Admin dashboard with analytics
- Bill payment integration (M-Pesa)
- Announcement broadcast system
- Email/SMS notifications
- Landlord portal
- Document management
- Community events calendar
- Visitor management

🛠️ Technology Stack

Component	Technology
Backend	Django 5.2 (Python)
Database	SQLite (Development)
Frontend	HTML5, CSS3, JavaScript
Icons	Font Awesome 6.4
Version Control	Git & GitHub
IDE	Visual Studio Code

📁 Project Structure

```
Jirani App/
├── dashboard/          # Main Django app
│   ├── migrations/     # Database migrations
│   ├── static/          # Static files (CSS, JS, images)
│   │   ├── css/
│   │   │   └── style.css  # Main stylesheet
│   │   ├── js/
│   │   │   └── main.js    # JavaScript functionality
│   │   └── images/       # Logo and images
│   ├── templates/       # HTML templates
│   │   └── dashboard/
│   │       ├── dashboard.html  # Main dashboard
│   │       └── login.html      # Login page
│   ├── models.py          # Database models
│   ├── views.py           # View functions
│   ├── urls.py            # URL routing
│   └── admin.py           # Admin configuration
├── jirani_project/        # Django project settings
│   ├── settings.py        # Project settings
│   ├── urls.py            # Main URL configuration
│   └── wsgi.py            # WSGI config
├── manage.py              # Django management script
├── db.sqlite3             # Database file
└── README.md              # This file
```

🚀 Installation & Setup

Prerequisites

- Python 3.11 or higher
- pip (Python package manager)
- Git

Step 1: Clone the Repository

bash

```bash
git clone https://github.com/YOUR-USERNAME/jirani-app.git
cd jirani-app
```

Step 2: Install Dependencies

bash

```bash
pip install django pillow
```

Step 3: Run Migrations

```bash
python manage.py makemigrations
python manage.py migrate
```

Step 4: Create Superuser

```bash
python manage.py createsuperuser
```

Step 5: Create Sample Data (Optional)

```bash
python manage.py shell
```

Then paste the sample data creation script (see DOCUMENTATION.md)

Step 6: Run the Server

```bash
python manage.py runserver
```

Visit: `http://127.0.0.1:8000/login/`

👤 Default Login Credentials

Resident Account:

- Username: `john.doe`
- Password: `password123`

Admin Account:

- Username: `admin`
- Password: [Your created password]
- Admin Panel: `http://127.0.0.1:8000/admin/`

📊 Database Models

Resident

- Links to Django User model
- Stores unit number, phone, move-in date
- Tracks active status

Payment

- Tracks rent, additional services, maintenance, debt
- Due dates and payment status
- Linked to residents

Request

- Service/maintenance requests
- Status tracking (pending, in progress, completed)
- Linked to residents

WorkOrder

- Work order management
- Priority and status tracking
- Assigned personnel

Unit

- Property unit information
- Status (vacant, occupied, upcoming)
- Optional images

Announcement

- Community announcements
- Timestamp and author tracking

🎨 Design System

Color Palette

```css
```

```
Primary (Indigo Blue): #667EEA
Secondary (Purple): #764BA2
Success (Green): #10B981
Warning (Amber): #F59E0B
Danger (Red): #EF4444
Dark (Charcoal): #1F2937
```

Typography

- Font Family: Segoe UI, Tahoma, Geneva, Verdana, sans-serif
- Headings: Bold (700-800 weight)
- Body: Regular (400 weight)

📸 Screenshots

Login Page

Beautiful gradient login interface with brand colors

Dashboard

- Left sidebar: Navigation and quick contacts
- Top bar: Page navigation and user profile
- Main content: Stats cards, charts, and tables
- Building overview with occupancy stats
- Payment summary cards
- New requests display
- Work orders tracking
- Vacant units showcase

🔐 Security Features

- CSRF protection on all forms
- User authentication required for dashboard
- Password hashing with Django's built-in system
- SQL injection prevention via Django ORM
- XSS protection through template escaping

📝 Development Timeline

Week 1-2: Setup, Models, Authentication
Week 3: Dashboard UI, Views
Week 4: Payment System, Requests
Week 5: Testing, Documentation, Deployment

🐛 Known Issues & Limitations

1. No actual payment gateway integration yet
2. Building image placeholder needed
3. Admin dashboard not yet implemented
4. No email notification system
5. Limited to one estate (no multi-tenancy)

🔮 Future Enhancements

☐ M-Pesa payment integration
☐ Email and SMS notifications
☐ PDF report generation
☐ Mobile app version
☐ Real-time chat support
☐ Visitor management system
☐ Document upload/storage
☐ Analytics dashboard for admin
☐ Multi-estate support

🤝 Contributing

This is an educational project. Suggestions and feedback are welcome!

1. Fork the repository
2. Create a feature branch
3. Commit your changes
4. Push to the branch
5. Open a Pull Request

📄 License

This project is developed for educational purposes as part of a school project.

🧑‍💻 Author

[Your Name]
Student at SCITT
[Your Email]
[LinkedIn Profile]

🙏 Acknowledgments

- Django Documentation
- Font Awesome for icons
- SCITT Institution
- [Lecturer Name]

📞 Support

For questions or issues:

- Create an issue on GitHub
- Email: [your.email@example.com]

⭐ **If you find this project helpful, please consider giving it a star on GitHub!**

Last Updated: November 2025